Technical Guide

# Lightweight Directory Access Protocol (LDAP)

## Feature Overview and Configuration Guide

## Introduction

Lightweight Directory Access Protocol (LDAP) is a software protocol used to manage and access various IT resources e.g. applications, servers, networking equipment, and file servers. The common use of LDAP is to provide a central place for authentication, meaning it stores usernames and passwords.

As the name suggests LDAP is a lightweight version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network. LDAP uses directories to maintain the organization information, personnel information, and resource information.

Network directories tell you where in the network something is located. On TCP/IP networks, the domain name system (DNS) is the directory system used to relate the domain name to a specific network address. However, if you don’t know the domain name, LDAP allows you to search for an individual without knowing where they're located.

You can use LDAP to authenticate users connecting to internal networks over OpenVPN. Although AlliedWare Plus devices can use both LDAP and RADIUS interchangeably as an authentication protocol, LDAP has the ability to interact with directory services such as Microsoft's Active Directory (AD). AD is one of the core pieces of Windows database environments. It stores user and account information, and provides authorization and authentication for computers, users, and groups, to enforce security policies across Windows operating systems.

This guide provides information for configuring OpenVPN Access Server to authenticate against Active Directory using LDAP.

# Contents



## Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support LDAP, running version **5.5.2-1** or later.

To see whether your product supports LDAP, see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

## Related documents

The following documents give more information about the authentication features on AlliedWare Plus products:

- the OpenVPN Feature Overview and Configuration Guide
- the AAA and Port Authentication Feature Overview and Configuration Guide
- The product's Command Reference

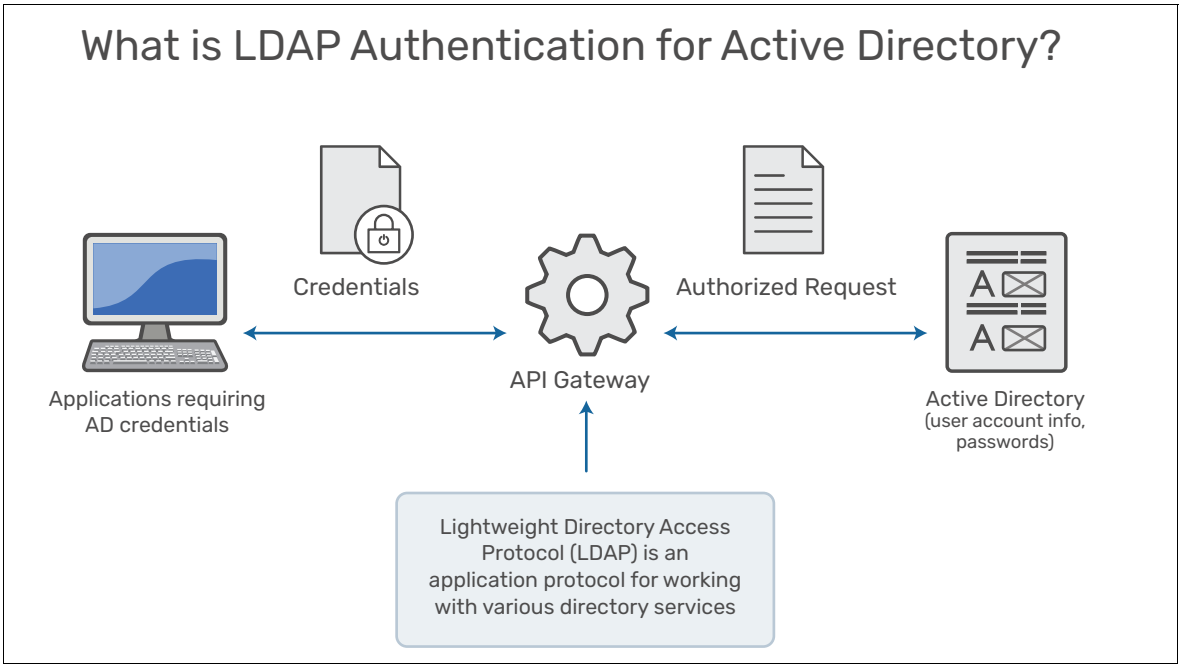
These documents are available from the links above or on our website at alliedtelesis.com

# LDAP overview

The LDAP protocol communicates with Active Directory. It's essentially a way to talk to Active Directory and transmit messages between AD and other parts of your network.

How does Active Directory LDAP authentication work? Basically, you need to set up LDAP to authenticate credentials against Active Directory. The 'BIND' operation is used to set the authentication state for an LDAP session in which the LDAP client connects to the server.

In most cases, this type of simple authentication essentially means a name and password are used to create a **bind** request to the server for authentication.



## Basic LDAP packet communication using Telnet

When you use Telnet to login to an AlliedWare Plus device, the basic LDAP authentication process is as follows:

1. Telnet initiates a connection request and sends a username and password to the device.
2. After receiving the request, the device (acting as the LDAP client), establishes a TCP connection with the LDAP server.

   To obtain the right to search, the device uses the administrator distinguished name (DN) and password to send an administrator **bind** request to the LDAP server.
3. The LDAP server processes the request. If the bind operation is successful, the LDAP server sends an acknowledgement to the device.
4. The device sends a user DN search request with the username to the LDAP server.
5. After receiving the request, the LDAP server searches for the user DN by the base DN, search scope, and filtering conditions. If a match is found, the LDAP server sends a response to notify the device of the successful search. There might be one or more user DNs found.
6. The device uses the obtained user DN and enters the user's password as parameters to send a user DN bind request to the LDAP server, which checks whether the user password is correct.
7. The LDAP server processes the request, and sends a response to notify the device of the bind operation result. If the bind operation fails, the device uses another obtained user DN as the parameter to send a user DN bind request to the LDAP server. This process continues until a DN is bound successfully or all DNs fail to be bound. If all user DNs fail to be bound, the device notifies the user of the login failure and denies the user's access request.
8. The device and server perform authorization exchanges.
9. After successful authorization, the device notifies the user of the successful login.

### Current AlliedWare Plus limitations

- Only one trustpoint is supported for secure LDAP.
- Recursive group searching is not implemented. However, with Active Directory, it is possible to set a specific OID as part of the search filter that will instruct it to perform a nested search.

The OID becomes part of the **memberOf** check:

```
memberOf:1.2.840.113556.1.4.1941:=<Group DN>
```

Rather than the usual:

```
memberOf=<Group DN>
```

There are examples of this in the search configuration section below, see "Search configuration" on page 10.

# Check list for logging in to an AlliedWare Plus device

Before you configure LDAP, login to an AlliedWare Plus device using SSH/Telnet, and check the following configurations are correct.

Check that:

1. An LDAP server is running.
2. You are able to reach the device, and the device can reach the LDAP server.
3. For the device:
   - SSH or Telnet is enabled
   - LDAP server is enabled
   - LDAP server is part of the AAA LDAP group server list
   - LDAP server group is added to the AAA login authentication options
   - LDAP server group is added to the vty lines login authentication options
4. For the LDAP server:
   - the following attributes are configured:

| LDAP Attribute | Format | Description |
|---|---|---|
| msRADIUSServiceType | Integer | To login to the AlliedWare Plus device, the user must have one of the following values:<br>■ 6 (Administrative): the user is mapped to the maximum user privilege, 15,<br>■ 7 (NAS Prompt): the user is mapped to the minimum user privilege, 1.<br>If this attribute is not configured or configured with different values, the user is not allowed to log in. |

## Network access via OpenVPN

To enable a user to connect to an internal network through OpenVPN, check that:

1. An LDAP server is running.
2. The user is able to reach the device, and the device can reach the LDAP server.
3. For the device:
   - LDAP server is enabled
   - LDAP server is part of the AAA LDAP group server list
   - LDAP server group is added to the OpenVPN AAA authentication options
   - An OpenVPN tunnel is configured and enabled

4. For the LDAP server:

   - the following user attributes are configured and passed to the OpenVPN client:

| LDAP Attribute | Format | Description |
|---|---|---|
| msRADIUSFramedIPAddress | Integer | Static IP address of the client. This is 4-byte integer. For example "-1062731519" is for "192.168.1.1". |
| msRADIUSFramedRoute | String | Static IP routes for the client (allows multiple entries). The string is expected to be in the format of the RADIUS attribute "Framed-Route" described in RFC2865, (e.g. "10.1.1.0 255.255.255.0 192.168.1.1 1") |
| ms-RADIUS-FramedIpv6Prefix | String | Static IPv6 prefix for the client. The string is expected to be in the format of "IPv6Address/PrefixLength", (e.g. "2001:1::/64"). |
| ms-RADIUS-FramedIpv6Route | String | Static IPv6 routes for the client (allows multiple entries). The string is expected to be in the format of the RADIUS attribute "Framed-IPv6-Route" described in RFC3162, (e.g. "3001:1::/64 2001:1::1 1"). |

# Configuring LDAP

This section describes how to configure LDAP, with some of the available AlliedWare Plus commands:

## LDAP server configuration

### Step 1: Create an LDAP server with the name AD_server

```
awplus#configure terminal
awplus(config)# ldap-server AD_server
```

### Step 2: Configure an IP address on the LDAP server

```
awplus(config-ldap-server)# host 192.0.2.1
```

### Step 3: Set the default base DN to use for searches

```
awplus(config-ldap-server)# base-dn dc=foo,dc=bar
```

### Step 4: Set the distinguished name with which to bind to the server and the credentials with which to bind

```
awplus(config-ldap-server)# bind authentication root-dn cn=Administrator,
cn=Users,dc=foo,dc=bar password P@ssw0rd
```

## AAA configuration

### Step 1: Create an LDAP server group called ldapServerGroup

```
awplus(config)# aaa group server ldap ldapServerGroup
```

Alternatively, you can use the default group ‘ldap’ which contains all LDAP servers.

### Step 2: Add the LDAP server AD_server to the group

```
awplus(config-ldap-group)# server AD_server
```

### Step 3: Create an AAA login method using the LDAP server group for user login authentication

```
awplus(config)# aaa authentication login ldapLogin group ldapServerGroup
```

Or use the default group instead:

```
awplus(config)# aaa authentication login ldapLogin group ldap
```

## SSH/Telnet configuration

### Step 1: Enable SSH

```
awplus(config)# service ssh
```

Make sure the SSH server is properly configured for users to login.

```
awplus(config)# ssh server allow-users userA
```

Step 2: **Authenticate VTY lines with the AAA authentication method ldapLogin**

```
awplus(config)# line vty 0 3
awplus(config-line)# login authentication ldapLogin
```

## OpenVPN configuration

Step 1: **Enable LDAP authentication of OpenVPN tunnels globally**

Again, you can either use the default LDAP group or a user defined LDAP group.

```
awplus(config)# aaa authentication openvpn default group ldap
```

# Secure Mode configuration - LDAPS using TLS encryption

LDAP offers a secure mode called LDAPS, which uses the TLS protocol to encrypt all communications between the client and server. To use LDAP, you must configure a secure port on the server (the default port is 636).

Once LDAPS has been configured on the server side, you will need a copy of the CA certificate used by the server. First this certificate needs to be imported onto the device in the form of a secure trustpoint. For more information on PKI and trustpoints on AlliedWare Plus, see the PKI Feature Overview and Configuration Guide.

Step 1: **Create a new PKI trustpoint called AD_trustpoint**

```
awplus(config)# crypto pki trustpoint AD_trustpoint
```

Step 2: **Specify that this trustpoint will use an external certificate that is copy and pasted into the terminal**

```
awplus(ca-trustpoint)# enrollment terminal
```

Step 3: **Return to privileged EXEC mode**

```
awplus(ca-trustpoint)# end
```

Step 4: **Import the external certificate to the trustpoint**

```
awplus# crypto pki authentication AD_trustpoint
```

The system will prompt for the certificate to be pasted into the terminal, in PEM format. Copy and paste the certificate.

```
Paste the certificate PEM file into the terminal.
Type "abort" to cancel.
```

Check the fingerprint and issuer information, and if everything looks correct, accept the certificate.

```
The certificate has been validated successfully.
Accept this certificate? (y/n): y
```

Step 5: **After accepting the certificate, return to the configuration terminal**

```
awplus#configure terminal
```

Step 6: **Enter configuration mode for the LDAP server name AD_server**

```
awplus(config)# ldap-server AD_server
```

Step 7: **Set the LDAP server hostname**

For Secure Mode, you will need to use a FQDN as the hostname, and this must match the name on the CA certificate you imported before. The LDAP server will perform a name check to ensure these names match, before the TLS session can be initiated.

```
awplus(config-ldap-server)# host example-FQDN.com
```

Step 8: **Enable LDAPS with TLS**

```
awplus(config-ldap-server)# secure mode
```

Step 9: **Add the LDAP server trustpoint created above**

```
awplus(config-ldap-server)# secure trustpoint AD_trustpoint
```

Step 10: **Optionally, specify the ciphers to use for TLS**

```
awplus(config-ldap-server)# secure cipher DHE-DSS-AES256-GCM-SHA384
AES128-GCM-SHA256
```

# Server connection configuration

Step 1: **Timeout setting**

When connecting to the directory server and when waiting for searches to be complete, the maximum wait time is 50 seconds.

```
awplus(config-ldap-server)# timeout 50
```

Step 2: **Retries**

When connecting to active servers, attempt 5 retries maximum.

```
awplus(config-ldap-server)# retransmit 5
```

Step 3: **Deadtime**

The device will not send any requests to the server for 5 minutes if it has failed to respond to a previous request.

```
awplus(config-ldap-server)# deadtime 5
```

# Search configuration

### Step 1: **Group DN settings**

For user authentication to be successful, the user must belong to the group with the Distinguished Name (DN) string: cn=Users,dc=test. By default it will determine this by checking the uniquemember attribute of the group, to see if it contains the users DN string.

```
awplus(config-ldap-server)# group-dn cn=Users,dc=test
```

### Step 2: **Active Directory group-attribute member settings**

For Active Directory, you will instead want to check within the member attribute of the group, which can be configured with the group-attribute CLI.

```
awplus(config-ldap-server)# group-attribute member
```

With those two options configured, a search would test a user's membership of group cn=Users,dc=test by checking the member attribute for the user's DN. This is useful when an LDAP server provides authentication information to a pool of clients, but the device should authorize only on a group of users.

### Step 3: **Login username settings**

The login name will belong to the attribute ‘username’. For user authentication to be successful, the directory must have an entry with username=<user>, e.g. username=jdoe.

```
awplus(config-ldap-server)# group-attribute username
```

### Step 4: **Search filter settings**

When retrieving user information, the users objectclass must contain, for example, ‘testAccount’ for user authentication to be successful. The search-filter option is highly customizable, and can be used to check any attribute. Additionally, boolean operators can be used to further enhance the specifics of the search.

```
awplus(config-ldap-server)# search-filter objectclass=testAccount
```

Examples:

- This would check that the users objectclass is testAccount OR organizationalRole

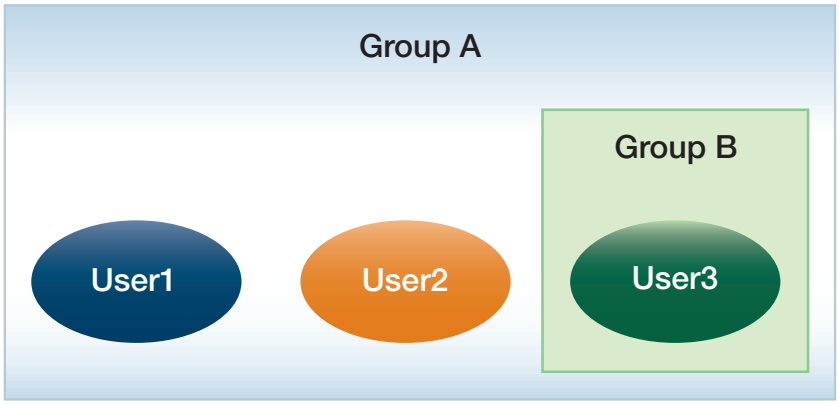
```
awplus(config-ldap-server)# search-filter
|(objectclass=testAccount)(objectclass=organizationalRole)
```

- This would check anyone who is a user AND NOT a computer

```
awplus(config-ldap-server)# search-filter
&(objectclass=user)(!(objectClass=computer))
```

## How to perform a nested search on Active Directory

Consider the following example:




- Without a nested search - using this search-filter below, any of the users within groupA will be able to login successfully, but user3 within groupB will fail.

```
awplus(config-ldap-server)# search-filter
memberOf=CN=groupA,OU=exampleOrg,DC=example,DC=test
```

- By adding the OID 1.2.840.113556.1.4.1941 into the memberOf check in the search-filter, Active Directory will recursively check all groups within groupA for the specified user. Now any users within any groups that are part of groupA will be checked, so our user3 in the nested groupB can login.

```
awplus(config-ldap-server)# search-filter
memberOf:1.2.840.113556.1.4.1941:=CN=groupA,OU=exampleOrg,DC=example,DC=test
```

# Monitoring LDAP

The following section provides some example output from the command **show ldap server group**. The output shows that there are two LDAP servers: Server_A and Server_B.

For Server_A, the show command specifies that:

- Server_A is alive
- Server_A is a LDAP server
- Server_A is part of the server group Management

For Server_B, the show command specifies that:

- Server_B is never used or the state is unknown.
- Server_B is an LDAP server
- Server_B is not part of any server group

There is also a second server group ‘RandD’ that does not have any LDAP servers.

```
awplus#show ldap server group
LDAP Group Configuration
  Group Name : ldap
    LDAP server name     Server Host/IP Address          Port  Status
    ---------------------------------------------------------------------
    Server_A             192.0.2.1                      N/A   Alive
    Server_B             192.0.2.2                      N/A   Unknown

  Group Name : Management
    LDAP server name     Server Host/IP Address          Port  Status
    ---------------------------------------------------------------------
    Server_A             192.0.2.1                      N/A   Alive

  Group Name : RandD
    LDAP server name     Server Host/IP Address          Port  Status
    ---------------------------------------------------------------------
    No LDAP servers currently defined
```

The two other server status states not shown in our example output are:

- Dead - the server is detected as dead and it will not be used for deadtime period.
- Error - The server is not responding.

## Enabling debug

As LDAP is configured under the AAA subsystem, the existing debugging for AAA authentication will generate useful information of LDAP operation.

```
awplus# debug aaa authentication
```

For detailed LDAP client debugging, with various debugging options, use the command:

```
awplus# debug ldap client
```

Note that turning on all LDAP client debugging may affect the system performance with a huge amount of log messages.